

Breviar

Laborator 1 - Introducere în Java

- Responsabil: Mihai Nan [mailto:mihai.nan.cti@gmail.com]
- Profesor titular: Carmen Odubășteanu

Introducere

Programarea Orientată pe Obiecte este o paradigmă de programare care utilizează obiecte și interacțiuni între acestea pentru a modela arhitectura unui program.

Aceasta constă în identificarea unor obiecte cu operații (metode) specifice asociate și realizarea comunicării între aceste obiecte prin intermediul unor mesaje. Elementul constructiv - **obiectul** - este o instanță a unei **clase** (tip definit de utilizator), iar clasele sunt membre ale unei ierarhii, fiind corelate între ele prin relații de **moștenire**.

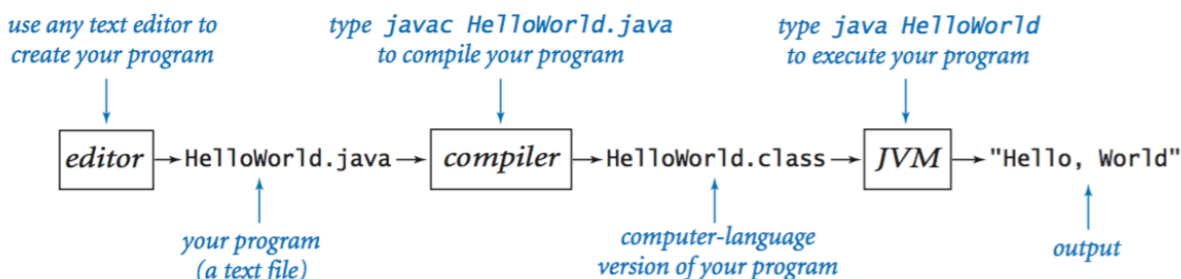
Folosirea **POO** permite realizarea de sisteme informatice de dimensiuni mărite, cu timpi de dezvoltare, testare și mentenanță reduși față de paradigmele anterioare. Totuși, pentru a crea un sistem funcțional este necesară înțelegerea corectă a conceptelor care stau la baza acestui tip de programare, iar de acest aspect se vor ocupa cursul și laboratoarele de la această disciplină.

Platforma Java

Java este un limbaj de programare orientat-obiect, puternic tipizat, conceput de către James Gosling la **Sun Microsystems** la începutul anilor '90, fiind lansat în 1995. Cele mai multe aplicații distribuite sunt scrise în **Java**, iar noile evoluții tehnologice permit utilizarea sa și pe dispozitive mobile. În felul acesta, se creează o platformă unică, la nivelul programatorului, deasupra unui mediu eterogen extrem de diversificat. De asemenea, acest limbaj este utilizat, în prezent, cu succes și pentru programarea aplicațiilor destinate intranet-urilor.

Limbajul împrumută o mare parte din sintaxă de la **C** și **C++**, dar are un model al obiectelor mai simplu și prezintă mai puține facilități de nivel jos. Un program Java compilat, corect scris, poate fi rulat fără modificări pe orice platformă pe care e instalată o mașină virtuală Java (în engleză **Java Virtual Machine**, prescurtat **JVM**). Acest nivel de portabilitate (inexistent pentru limbaje mai vechi cum ar fi C) este posibil deoarece sursele Java sunt compilate într-un format standard numit cod de octeți (în engleză byte-code) care este intermediar între codul mașină (dependent de tipul calculatorului) și codul sursă.

Mașina virtuală Java este mediul în care se execută programele Java. În prezent, există mai mulți furnizori de **JVM**, printre care **Oracle**, **IBM**, **Bea**, **FSF**. În 2006, **Sun** a anunțat că face disponibilă varianta sa de **JVM** ca open-source.

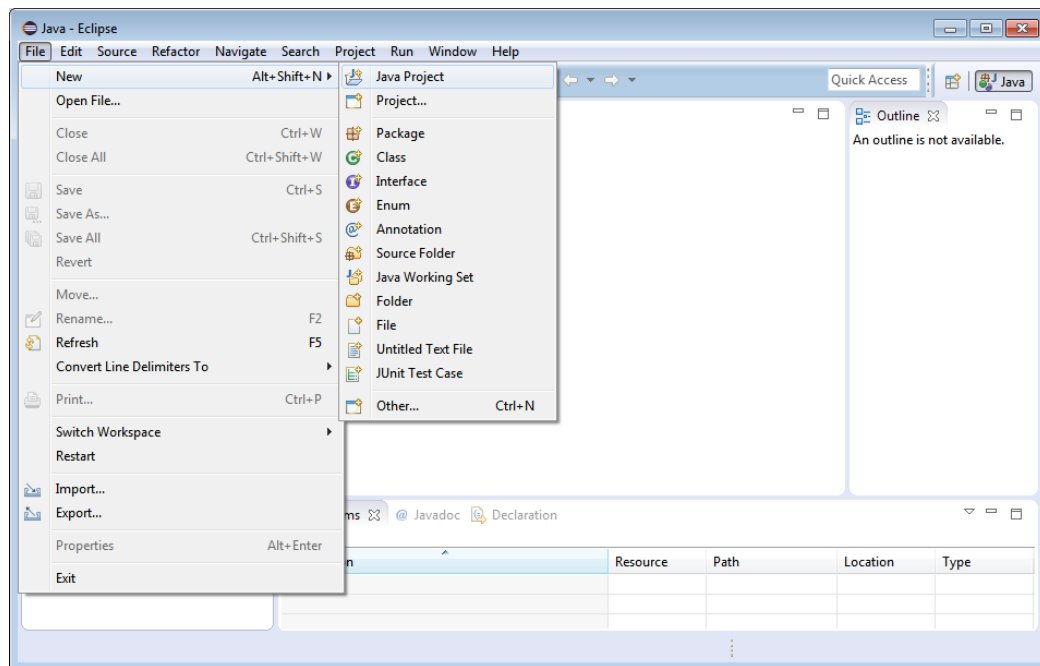


Mediul integrat Eclipse

Scrierea și rularea programelor Java

Pentru scrierea și rularea unei aplicații Java, utilizând acest IDE, se vor executa următorii pași:

1. Se va alege din meniul **File** opțiunea **New** și apoi **Java Project**.



2. Se va introduce numele proiectului - **Project Name**.
3. La alegere, se poate schimba locația unde se va salva proiectul(prin modificarea câmpurilor **Location**), precum și versiunea de JRE, recomandandu-se JavaSE-1.7 sau 1.8.

New Java Project

Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: HelloWorld

☒ Use default location

Location: U:\programs\Eclipse_44_Juna\HelloWorld [Browse...](#)

JRE

☒ Use an execution environment JRE: JavaSE-1.8

☐ Use a project specific JRE: jdk1.8.0_20 [Configure JREs...](#)

☐ Use default JRE (currently 'jdk1.8.0_20')

Project layout

☒ Use project folder as root for sources and class files [Configure default...](#)

☐ Create separate folders for sources and class files

Working sets

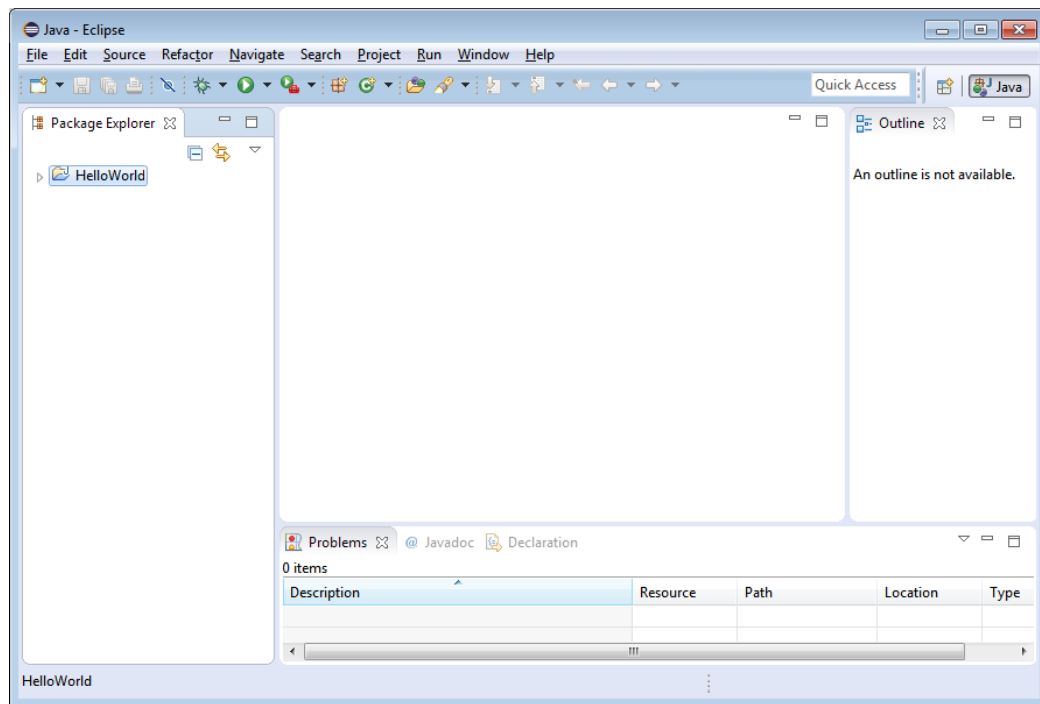
☐ Add project to working sets

Working sets: [Select...](#)

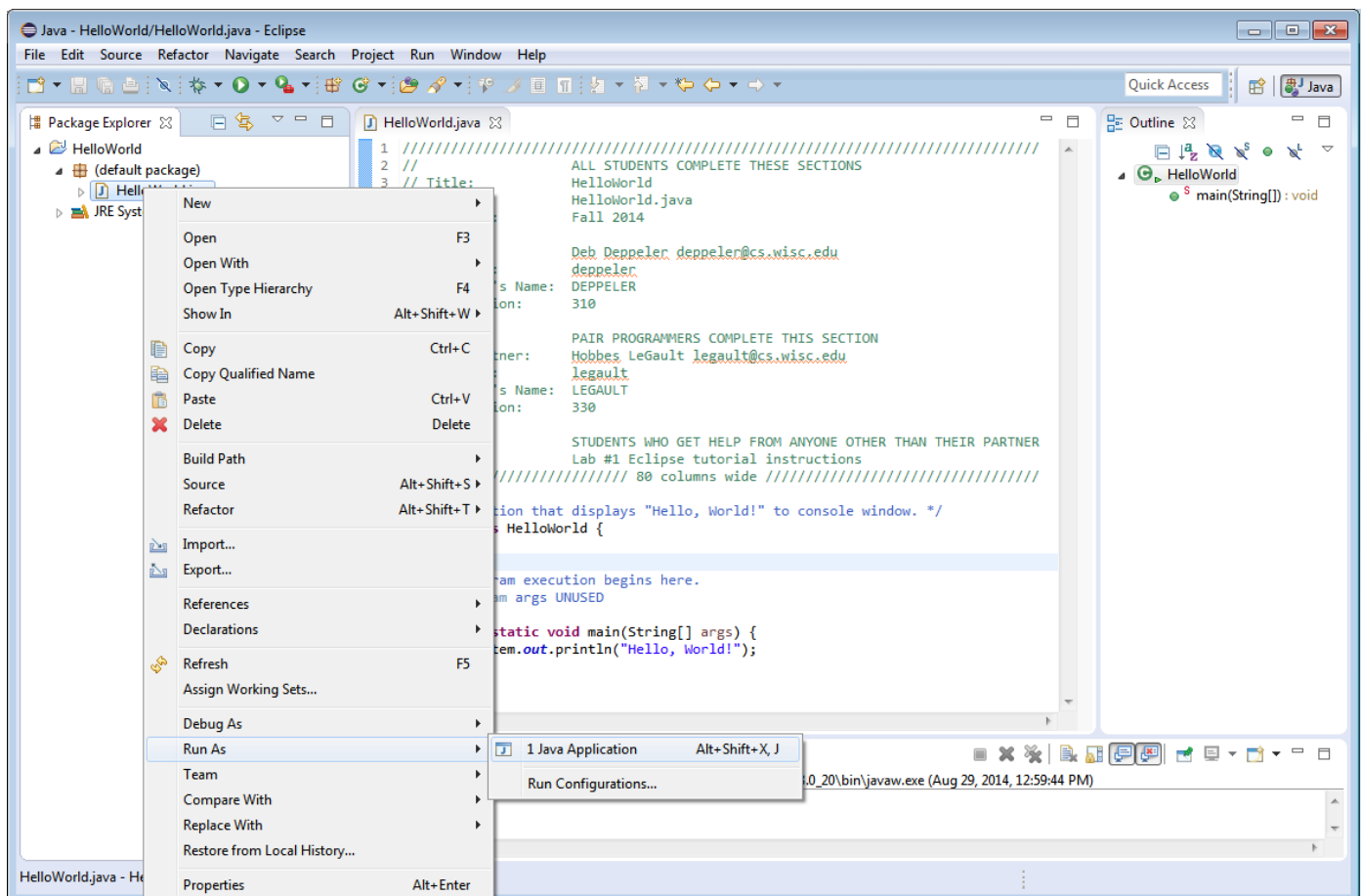
Information: The default compiler compliance level for the current workspace is 1.7. The new project will use a project specific compiler compliance level of 1.8.

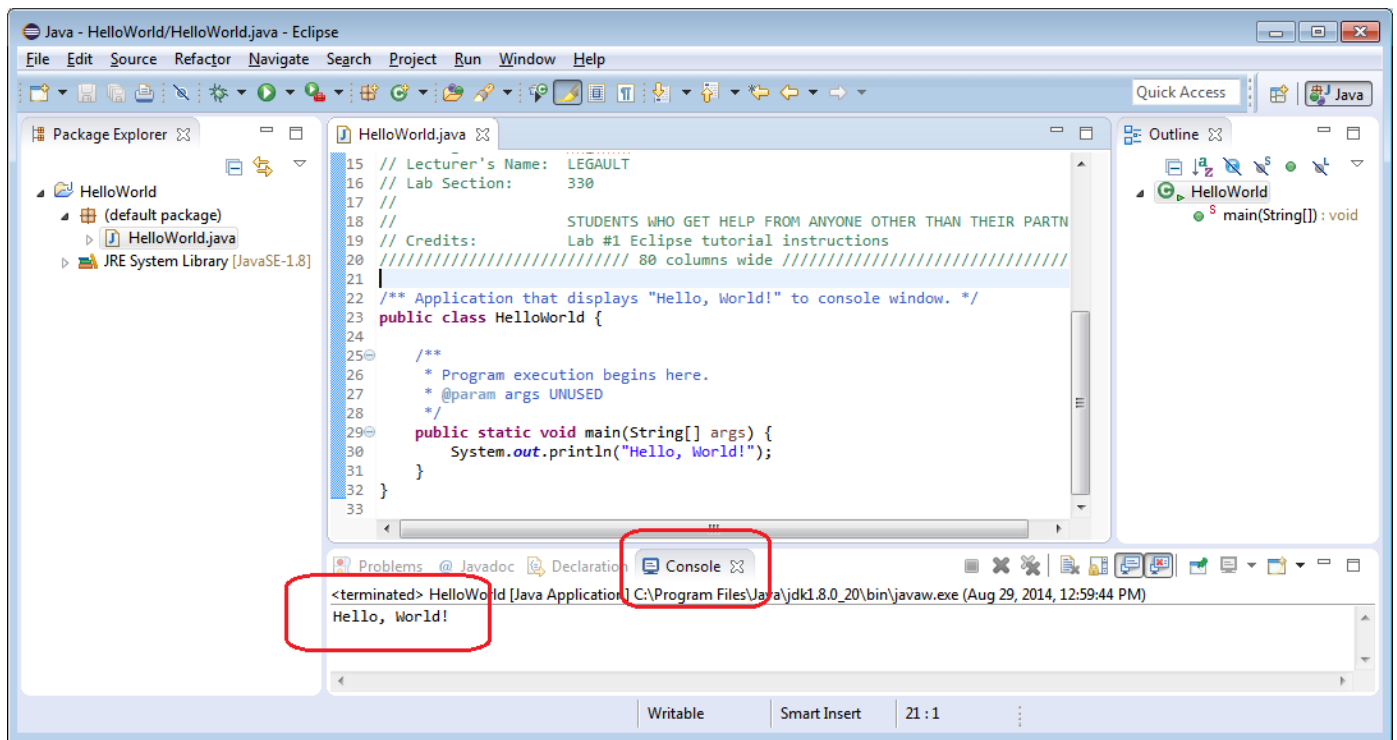
[?](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)

4. După ce proiectul a fost creat, se vor adăuga fișiere sursă după preferințe. Acest lucru se face selectând, cu click dreapta, numele proiectului din meniul **Package Explorer**, apoi se selectează **New..** și se alege **Class**. Pentru fișierul sursă ce va conține metoda main se poate bifa opțiunea **"public static void main(String[] args)"** pentru ca aceasta să fie adăugată automat. De asemenea, se poate introduce numele pachetului în care va fi adăugată sursa, în caz contrar, aceasta fiind pusă într-un pachet "default".



5. După introducerea numelui și apăsarea butonului **Finish**, în IDE se va deschide fișierul sursă nou creat.
6. Pentru Build, se va apăsa **CTRL + B**.
7. Pentru rulare, se va apăsa **CTRL + F11** sau se va selecta, din IDE, butonul având simbol un triunghi verde.





În **Eclipse** nu se pot compila și rula fișiere de sine stătătoare. Acestea trebuie să facă obligatoriu parte dintr-un proiect.

JDK Javadoc

Se va download documentația JDK de pe site-ul oficial Oracle [<http://docs.oracle.com/javase/7/docs/>]. Se downloadează din stânga JDK 7 Documentation. Apoi, în IDE, efectuați următorii pași:

1. Urmăți calea în meniul **Window** → **Preferences** → **Java** → **Installed JREs Edit**.
2. Se selectează **resources.jar**, **rt.jar**, **jsse.jar**, **jce.jar** și **charsets.jar**.
3. Apoi, se apasă **Javadoc Location..** și se completează calea către directorul **"/docs/api/"** din fișierele extrase din arhiva downloadată.

Parametri în linia de comandă

Pentru a folosi parametri în linia de comandă, se vor efectua următorii pași:

1. Se va executa un click dreapta pe numele proiectului și se va alege **Properties** din meniul pop-up.
2. Se va selecta ramura **Run** din arborele de configurare al proiectului și în câmpul **Arguments** se vor scrie parametrii doriți.

Mediul integrat NetBeans

Scrierea și rularea programelor Java

Pentru scrierea și rularea unei aplicații Java, utilizând acest IDE, se vor executa următorii pași:

1. Se va alege din meniul **File** opțiunea **New Project**.
2. Se va selecta **Java** din panoul **Categories**, respectiv **Java Application** din panoul **Projects**.
3. După apăsarea butonului **Next**, se va introduce numele proiectului - **Project Name**.
4. La alegere, se poate schimba locația unde se va salva proiectul, precum și directorul de lucru (prin modificarea câmpurilor **Project Location**, respectiv **Project Folder**).
5. Se va bifa căsuța **Set as Main Project** și se va debifa casuța **Create Main Class**. În acest fel, clasa care va conține metoda **main()** va avea același nume cu cel al proiectului.
6. După ce proiectul a fost creat, se vor adăuga fișiere sursă după preferințe. Acest lucru se face selectând, cu click dreapta, **Source Packages**, apoi din meniul **New** se va alege **Java Class**.
7. După introducerea numelui și apăsarea butonului **Finish**, în IDE se va deschide fișierul sursă nou creat.
8. Pentru compilare, se va apăsa tasta **F1** sau se va selecta, din IDE, butonul având ca simbol un ciocan.
9. Pentru rulare, se va apăsa tasta **F6** sau se va selecta, din IDE, butonul având simbol un triunghi verde.

În **NetBeans** nu se pot compila și rula fișiere de sine stătătoare. Acestea trebuie să facă obligatoriu parte dintr-un proiect.

JDK Javadoc

Se va download documentația JDK de pe site-ul oficial Oracle

[<http://www.oracle.com/technetwork/java/javase/downloads/index.html>]. Apoi, în IDE, efectuați următorii pași:

1. Alegeți din meniul principal **Tools** opțiunea **Java Platforms**.
2. Selectați platforma la care vreți să adăugați documentația din panoul din stânga al ferestrei de dialog.
3. În tab-ul **Javadoc** apăsați butonul **Add ZIP/Folder** și apoi specificați locația fișierelor **Javadoc**.
4. Apăsați butonul **Close**. Restartați IDE-ul. În acest moment, puteți accesa documentația din meniul **Help**, alegând **Javadoc References** și indicând **Java Platform SE 8**.

Parametri în linia de comandă

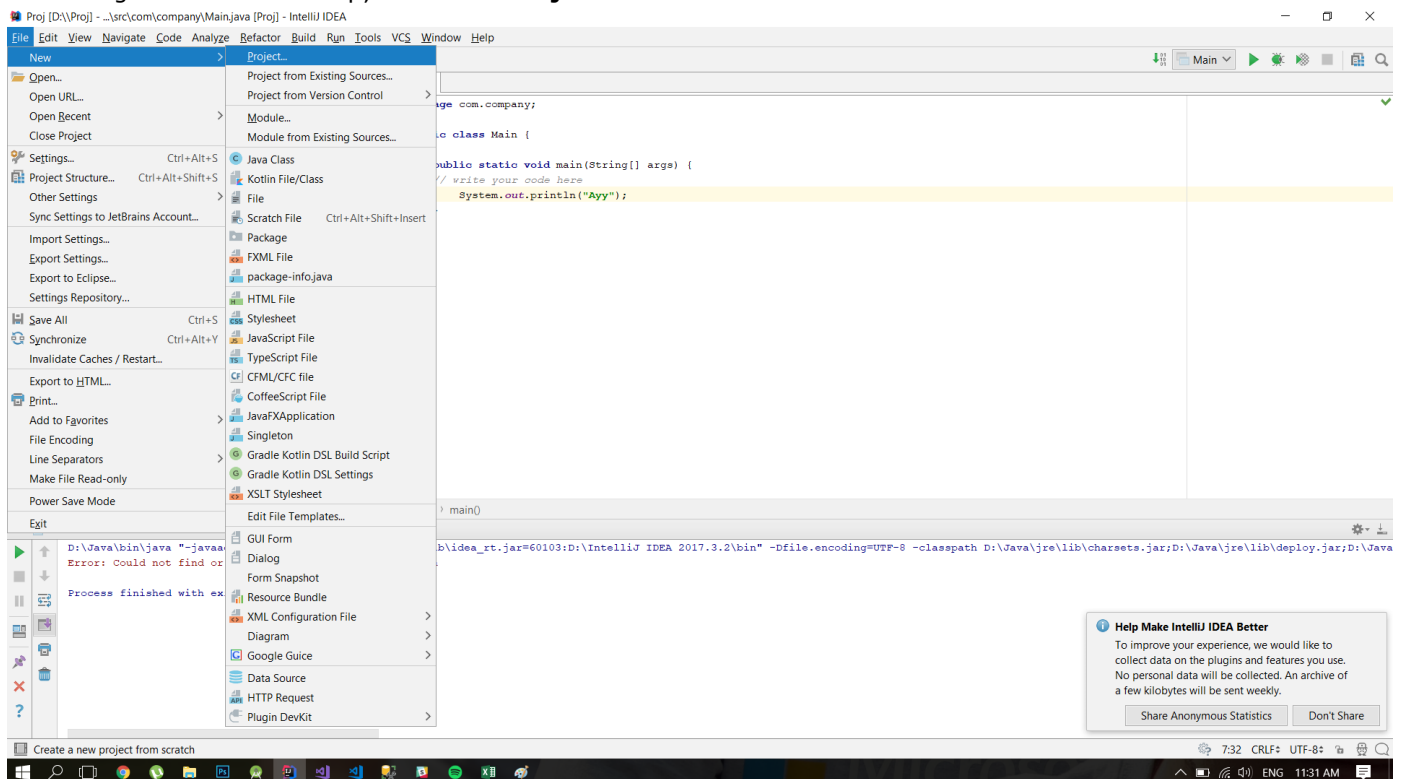
Pentru a folosi parametri în linia de comandă, se vor efectua următorii pași: 1. Se va executa un click dreapta pe numele proiectului și se va alege Properties din meniul pop-up. 2. Se va selecta ramura Run din arborele de configurare al proiectului și în câmpul Arguments se vor scrie parametrii doriți.

Mediul integrat IntelliJ IDEA

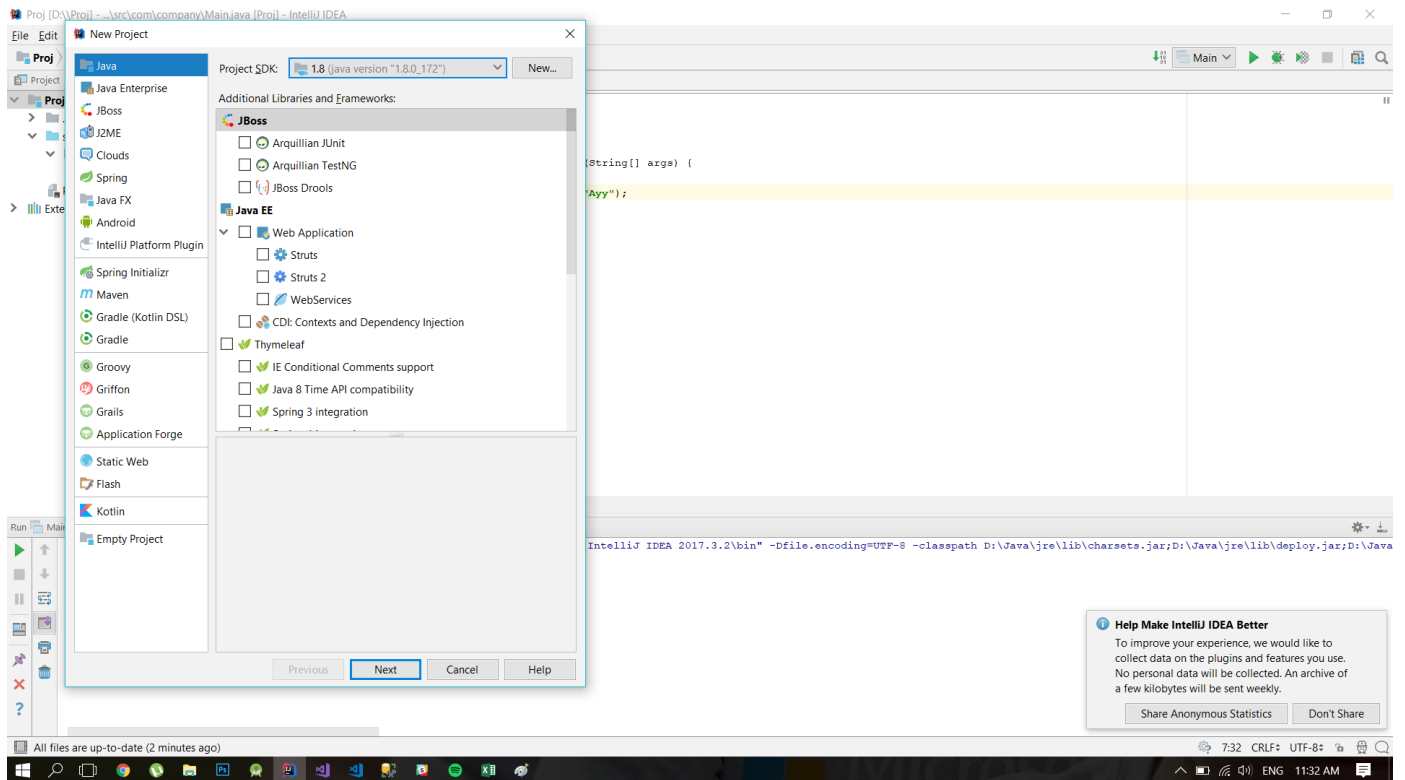
Scrierea și rularea programelor Java

Pentru scrierea și rularea unei aplicații Java, utilizând acest IDE, se vor executa următorii pași:

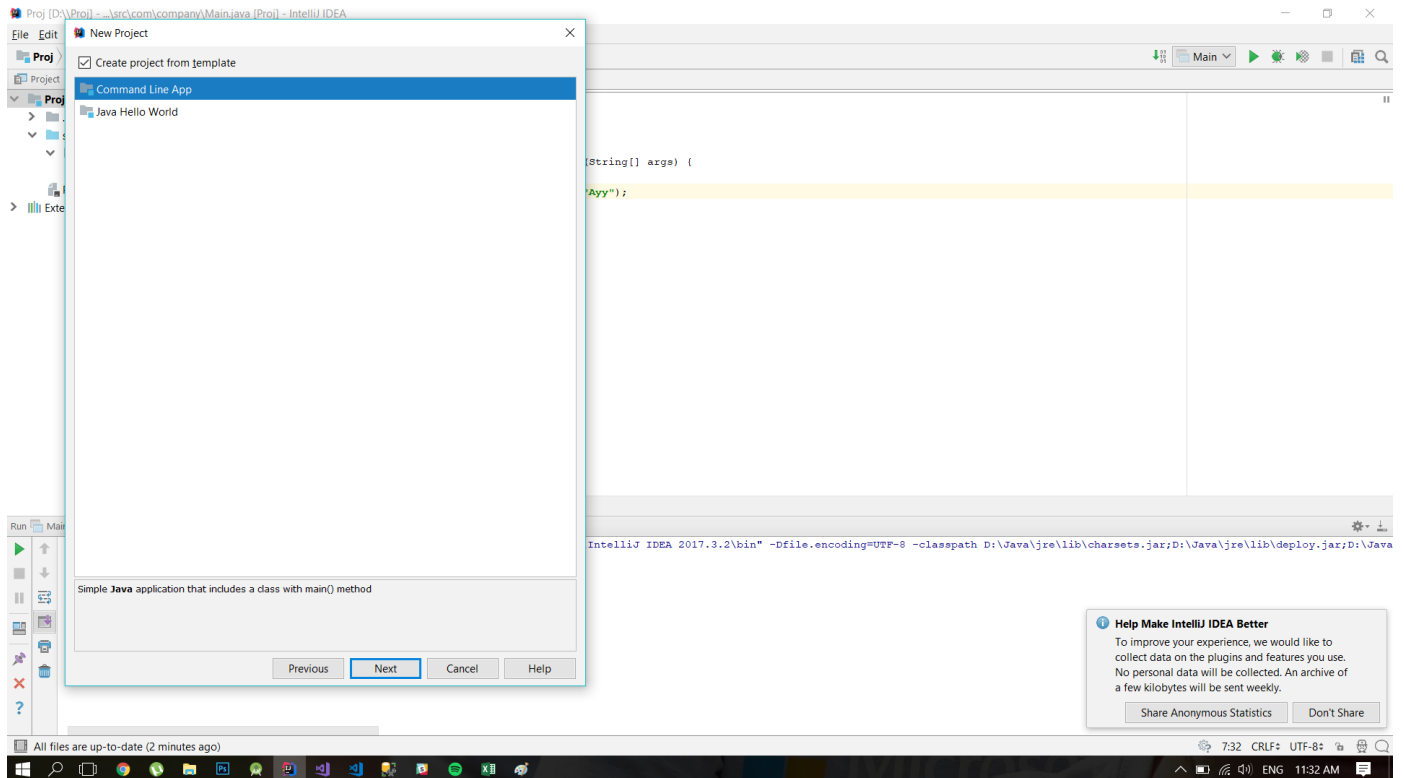
1. Se va alege din meniul **File** opțiunea **New Project**.



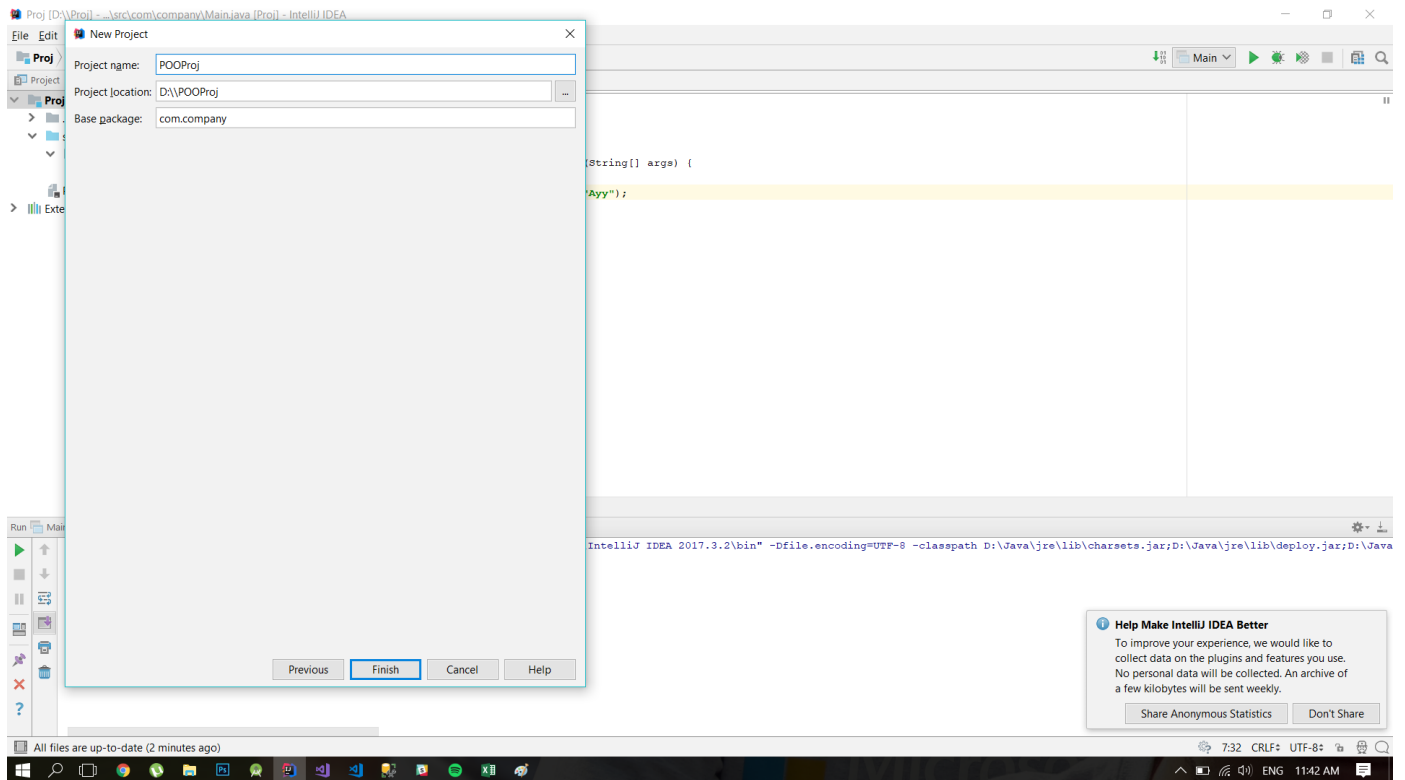
2. Se va alege **Java** din minifereastra din stânga (prima opțiune de sus) și puteți să alegeți versiunea de SDK pentru proiect, apoi dați next.



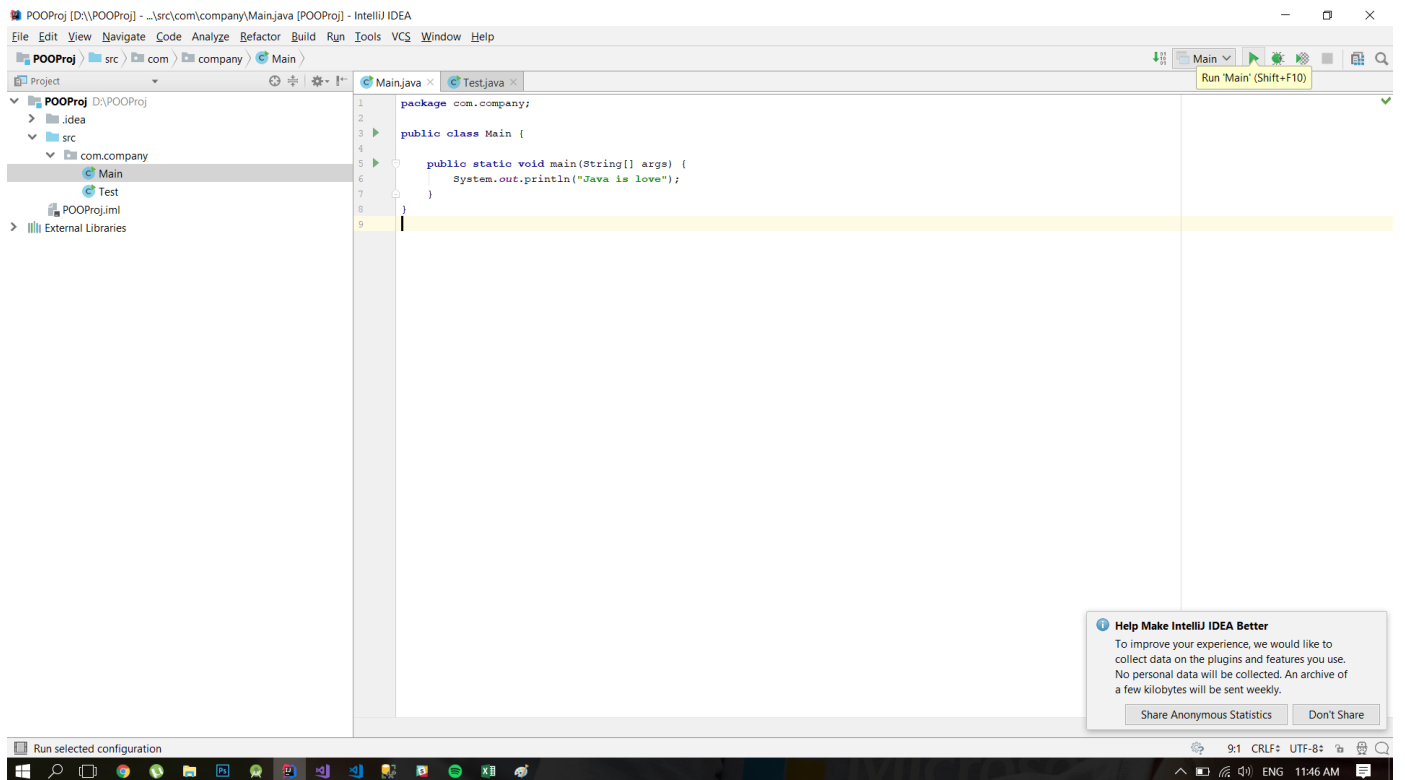
3. Va apărea o opțiunea **Create project from template**. O bifați pentru a selecta opțiunea de mai jos **Command Line App**, pentru a avea predefinită metoda main în clasa Main (care va fi de asemenea predefinită în proiect).



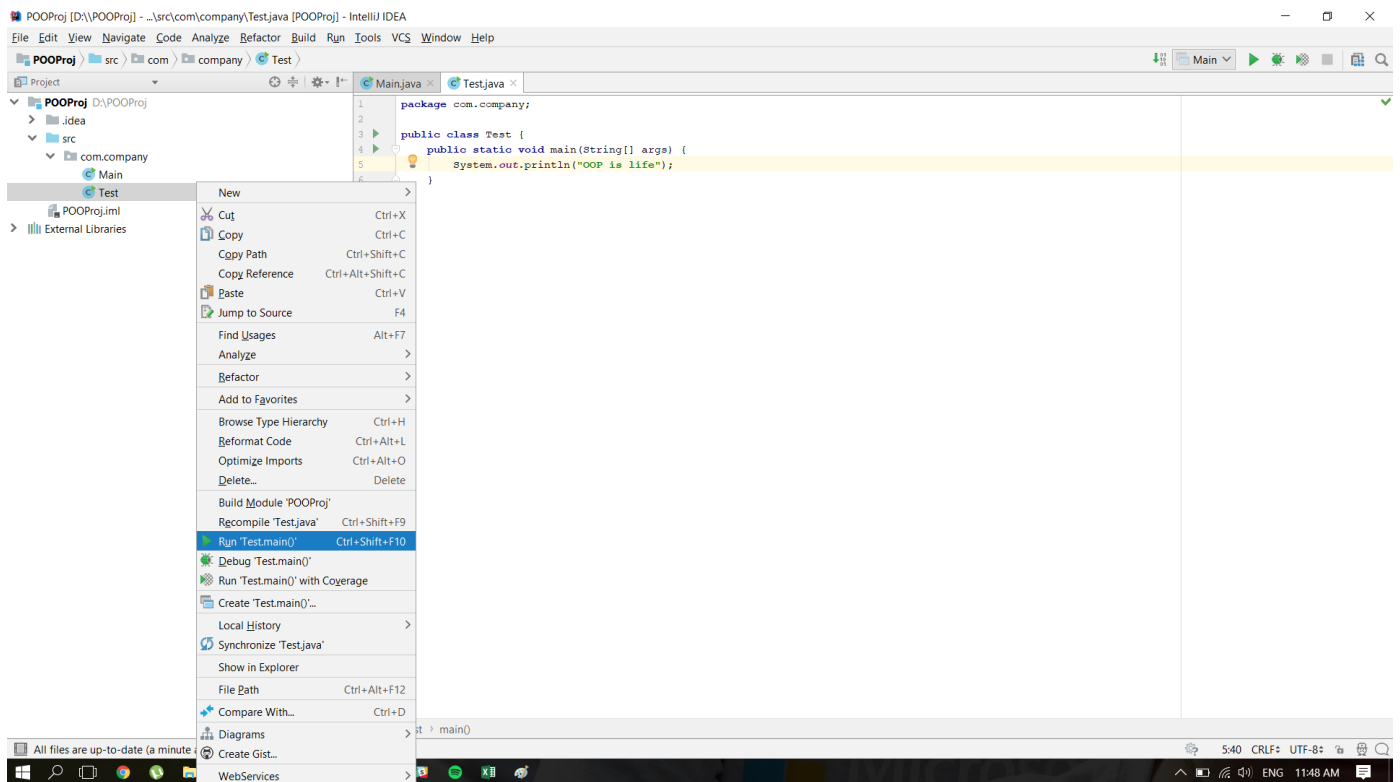
4. Adăugați un nume și o cale de acces pentru proiect.



5. În dreapta-sus, este un buton de rulare a programului (**Shift+F10**), iar pentru build puteți folosi shortcut-ul **Ctrl + F9**.



6. Într-un proiect puteți avea mai multe clase care conțin o metodă `main`, însă o singură clasă e folosită pentru rulare. Pentru a selecta o altă clasă ce conține metoda `main`, avem panoul din stânga cu ierarhia proiectului, click-dreapta pe clasa dorită și alegem opțiunea Run "**NumeClasa.main()**".

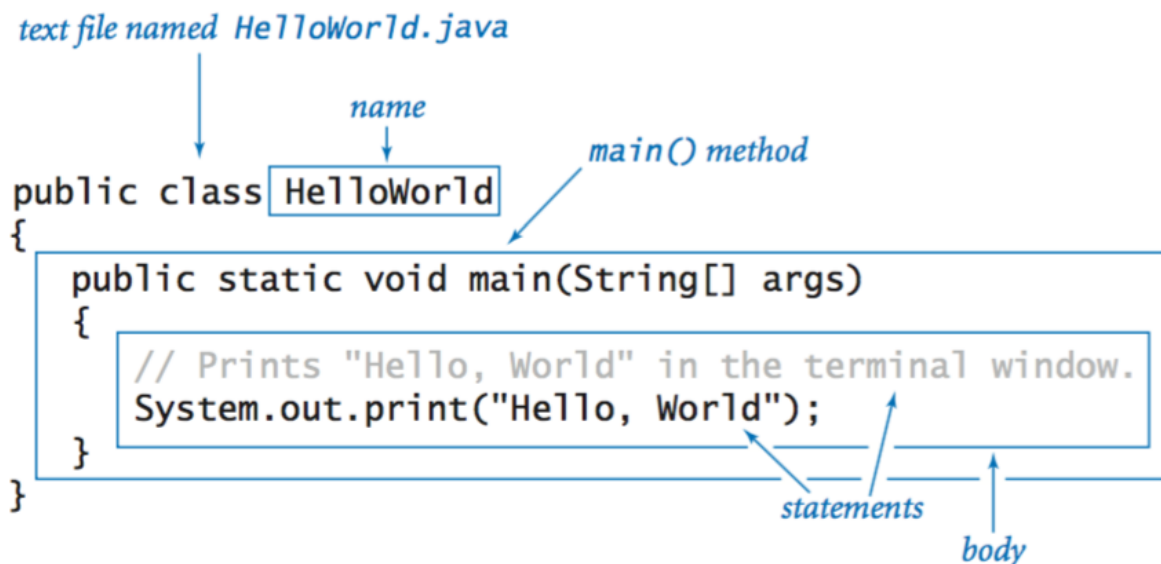


Parametri în linia de comandă

Pentru a folosi parametri în linia de comandă, se vor efectua următorii pași: 1. Se va alege comanda Run din meniul principal (orizontal) al IDE-ului 2. Se va selecta ramura Run din arborele de configurare al proiectului și apoi se da click pe sageata din dreptul clasei noastre 3. Se alege Edit și în câmpul Program arguments se vor scrie parametrii doriți.

Introducere în Java

Structura unui fișier



poo/breviare/breviar-01.txt · Last modified: 2021/10/11 14:52 by carmen.odubasteanu